EFFECTIVE HEURISTICS USED IN CELL SUPPRESSION CALCULATIONS

G. Sande
Gordon Sande, Sande and Associates, 600 Sanderling Court, Secaucus, New Jersey 07094, USA
g.sande@worldnet.att.net

ABSTRACT

The use of cell suppression to protect the confidentiality of respondent data in business statistics publications is common practice. This practice has been automated for some time. This note looks at the computational approximations and heuristics used as the basis of the automation. A good cell suppression pattern is obtained with of an optimization method applied to a figure of merit of proposed cell suppression patterns. Various figures of merit have been proposed which produce varying qualitative characteristics in the resulting cell suppression patterns. A number of deficiencies in the common greedy sequential heuristic can be identified in standard examples of cell suppression patterns. Modifications directed at reducing these deficiencies may be constructed.

Key Words: Statistical Disclosure Control, Statistical Disclosure Limitation, Statistical Confidentiality, Business Statistics

1. INTRODUCTION

The use of cell suppression to protect the confidentiality of respondent data in business statistics publications is common practice. This practice has been automated for some time (Robertson 1993). The purpose of this note is to look at the computational approximations and heuristics used as the basis of the automation. We assume that the general setup is known from other work so that we may focus on computational issues.

The typical business statistics publication is a cross tabulation with a low number, typically only two or three, of dimensions or classification variables but with each classification variable having a hierarchical structure of moderate depth. A typical *Standard Industrial Classification* (SIC) classification variable has four digit industries, three digit major industries, two digit industry groups, one digit industry divisions and a (no digit) grand total. A typical geographical classification variable may follow alternate hierarchical structures to allow either locality, county and region or locality, metropolitan area and region with regions grouped into states or provinces and these into a national total. The counties and metropolitan areas do not properly contain each other so it is technically inaccurate to describe this as a hierarchy, although we will abuse the terminology when the alternate hierarchical structures are available. With such classification variables, many tabulation cells are defined. We will describe these as either internal cells that are not further disaggregated or as marginal cells that may be disaggregated. Various other cells arise that are not part of the natural structure, which we will call miscellaneous aggregations. The use of two hierarchical classification variables will technically lead to a lattice of cell types. This restates the fact that a subtotal at the first level of disaggregation of the first classification variable may or may not be more or less disaggregated than a subtotal at the first level of disaggregation of the second classification variable as there is only a partial ordering under the notion of degree of disaggregation. There are often additional subtotals introduced to allow historical continuity or to allow some aspect of the subject matter to be represented that does not easily follow from the coding structure. Grouping of manufacturing into durable and nondurable goods is a common example that could be described as the ad hoc introduction of additional hierarchy into the classification variable. For the purposes of cell suppression we will want to know all cells that are logically defined. This may differ from the publication as it may not include some cells either because they are of low interest in an already bulky publication or they are part of highly disaggregated tables that would be mostly suppressed so little is lost by not publishing the few remaining cells.

Other miscellaneous aggregations will be purely of a technical nature, introduced as part of the definition of the cell suppression problem to allow for the presence of multiple establishment enterprises that contribute to multiple cells and for cells with very low enterprise counts. When the pooled sensitivity of two cells is close to the upper bound permitted by subadditivity we will represent the pooling explicitly with a miscellaneous aggregation. This allows the cell suppression computation to make the approximation that pooled sensitivities are close to the lower bound permitted by subadditivity. These technical miscellaneous aggregations will tend to be sensitive. Their components may be either sensitive cells or nonsensitive cells which will often be nearly sensitive. To further abuse the terminology, it is sometimes convenient to identify the naturally defined sensitive aggregations and exclude these technically defined sensitive aggregations.

The cell suppression problem is defined by this collection of cells, some of which are identified as being sensitive, and a set of relationships that define some of the cells as aggregations of other cells. The design of a suitable publication pattern is the final goal. We must define what we mean by suitable. The most immediate requirement is that there be no inadvertent residual disclosures, either exact or approximate, in the publication pattern.

2. BASIC FRAMEWORK

The basic mathematical framework is to view possible publications as points in a high dimensional space. There will be a special point which is the values that are listed internally where all of the values, sensitive or not, are shown including the cells that may not make it into the released publication. This point satisfies the defining

relationships between the cells and is a feasible point in mathematical programming terminology. Before the publication is released, the end users will have various estimates of the values of the cells. This is the prior knowledge that is used to construct the sensitivity criteria. The prior knowledge can be represented a collection of points in the publication space. The assumption is that this prior knowledge is not grossly in error and it is often treated as if it allowed the end user estimates to be between 50 and 150 percent of the true values. The detailed choices will yield varying parameters in the sensitivity criterion. This can be compactly represented as a polyhedron that represents the prior knowledge with the internal publication as a point in the interior of the polyhedron. When we publish some of the cell values, we provide values for some of the coordinates in the publication space that will act to restrict the consistent cell values to a polyhedron that is contained in the prior knowledge polyhedron. The released publication will not, if correctly done, permit the values for sensitive cells to be determined too accurately. The precise arithmetical meaning of too accurately will be specified as part of the sensitivity criterion. The design problem is to ensure that the publication is done correctly. Each sensitive cell will specify a collection of polyhedra that are the boundaries between too accurate and not too accurate. The specification is just a requirement on some coordinates in the publication space. Some member of this collection must be contained in the publication polyhedron. And this must be true for all the sensitive cells. Our interest will be in deviations about the internally known nominal value, so we will often use the polyhedron of deviations that is centered about zero rather than the publication polyhedron that is centered about the nominal values without commenting on this change of reference point.

To form a metaphor for this we can think of a box containing a nest of Russian dolls. The box is the prior knowledge polyhedron and the outer doll is the released publication polyhedron. Rather than a nest of dolls, we have a variety of inner dolls that must all fit inside the outer doll. The inner dolls are not really of fixed shape but can be described as bean bags that have a child's jack inside them. We want to design the outer doll to fit inside the box and to allow each of the bean bags to be contained in it.

3. OBJECTIVE FUNCTION

To use optimization techniques to design a publication pattern, we need an objective function to assign a figure of merit to possible patterns so that good, or even the best, patterns can be found. A common suggestion for an objective function for a cell suppression pattern is the count of cells suppressed. However, some observe that the cells are of differing size and should not be treated equally. The suggestion then becomes that the objective function should be the sum of the values of the cells suppressed. Both justifications are straight forward and quite similar. When ready justifications specify rather different objective functions it is safe to assume that we have just learned that no justification is likely to be convincing.

When we look at publications with suppressed cells we will see that not all suppressions are the same. We will have to decide whether it makes sense to consider a partially suppressed cell. In technical terms this is the question of whether we are to use integer variables to capture all of the cell value or to use continuous variables to capture part of the cell value. If a marginal cell is suppressed, not all of its disaggregates will also be suppressed. The remaining disaggregates provide a lower bound on the possible values that the marginal cell might have. The presence of defining relationships for marginal cells renders the justification for integer variables much less convincing. Without the defining relationships there would be no problem to solve and each suppression would automatically be all or nothing. There is also the practical consideration that integer variables make optimization problems harder to perform. One effect of the defining relations being only unweighted sums of cells is to cause the continuous computations to take on the same value with fluctuating sign in many simple cases. The difference between the two computational modes will often be rather small in the common simple cases, with the differences arising when simple cases are mixed together.

We can look at the use of disaggregates to estimate a lower bound for a marginal cell. Perhaps we should just apply our criteria to internal cells and take whatever that implies for the marginal cells. This will have the effect of suppressing many higher level aggregates and almost certainly the grand total which is the broadest of the aggregates. This is quite contrary to the design intent of a typical business survey and would not be considered acceptable. It does however keep bringing us back to the underlying question of what is a sensible figure of merit.

The figure of merit for a publication pattern is often called the information in the pattern. There are many information measures and a theory that characterizes them. For observed functions, such as the cell values, we would use the Berg entropy rather than the Shannon entropy used for probability distributions. The information of a cell would be the logarithm of the cell value. When we use the logarithm of the cell value as a coefficient in an objective function we may have negative coefficients. This can be avoided by the use of the started logarithm, $log(1+x)$. Logarithm is a common transformation for business data where it is often called ratio scaling.

Without a convincing prior argument we can always look at existing manual practice for suggestions of an objective function. Manual practice seems to seek two contradictory goals. There is a desire to avoid the suppression of large cells. This reflects the aspect of subject matter expertise of knowing which things are big. There is a desire to suppress only a few complements. This reflects the aspect of subject matter expertise of knowing the patterns in the data of which things are small, tiny or absent. For the cell suppression problem the absent, or empty, cells are known values of zero. The practice seems to be a balance between the small number of large cells that counting would yield and the many small cells that summing values would yield. Trials with counting suppressions, the Berg entropy or summing the values of suppressions tend to choose the middle ground as an acceptable balance.

4. GREEDY SEQUENTIAL HEURISTIC

We have a problem with a number of cells, some of which have been specified as being sensitive, and an objective function that is a figure of merit for proposed suppression patterns. For each cell we will have a nominal value

about which there are departures. The nominal values will satisfy a number of defining relationships, as will the departures. The departures will be x_i restricted to lie between numerically specified bounds, that would typically be *-1/2* to *+1/2* of the nominal value. We may collect this into the form $D\,x = 0$ with $l \le x \le u$ where the matrix D represents the various defining relationships with coefficients of *0*, *1* and *-1*. This form with an equality relationship in the equations and lower and upper bounds on the variables arises naturally here. It called the standard form and is commonly used in linear programming codes. These conditions define the prior knowledge polyhedron. To protect a *single* sensitive cell we impose the condition that the departure for that cell be above the restricted range. The problem definition is symmetric about the nominal values, so we could equally well have imposed the restriction that the departure would be below the restricted range. We would have $l_i \le x_i \le u_i$ for a new value of l_i. To satisfy the defining relations some other x_j will be nonzero and we would say that those other cells are to be suppressed. There are many possible choices of the other cells. We would choose to minimize the objective $c'|x|$. To deal with the absolute values $|x|$ we would either use an optimization code that does l_1 fitting or use the representation of a general variable as the difference of two positive variables. The objective function coefficients for the sensitive cells would be set to zero as we already know that those cells can not be published. Other cells that are known to be suppressed or otherwise absent from the publication would also be given a zero coefficient.

This will protect a *single* chosen sensitive cell. We can protect all the cells by choosing each one in turn. This is the basis for the usage *sequential* in the name. After any cell is protected, we will have a more extensive list of known suppressed cells for the next sensitive cell to be protected. This will allow cells chosen later to benefit from the complements that have already been chosen. If we examine the departures used to protect any particular sensitive cell, we may notice that they are adequate to protect some other sensitive cell. We would say that such a sensitive cell has been protected *in passing* and we would not need to choose it for explicit protection. If the initial cell chosen is the most sensitive cell and the cells chosen in turn are the most sensitive cells not yet protected, we would be following a common strategy usually known as the *greedy algorithm*. The combination leads to the name of *greedy sequential heuristic*.

Such an algorithm was implemented in the Statistics Canada Confidentiality Studies Software (CONFID) research prototype automated cell suppression system in 1979 and has been in use since then (Sande 1984). It is also the basis of the later versions of the USBC automated cell suppression system (Kirkendall and Sande 1998). The greedy sequential heuristic arises naturally in many problems. For most problems it provides a sensible starting point and for many it is even a sensible solution. For a limited number of special problems it can be proven to provide an optimal solution. It is not optimal for the knapsack problem which implies that it can not be optimal for the cell suppression problem.

In terms of our metaphor of nested Russian dolls, we first fit the outer doll around the largest bean bag to get an initial shape. In the process we fix the shape of the first bean bag. We now expand the outer doll to fit the next largest bean bag. Its shape will be partially determined by the outer doll's accommodation of earlier bean bags and partially by the jack that it contains. We continue through the bean bags in turn. Some will already fit and some will require expansion of the outer doll.

5. DEFECTS OF THE GREEDY SEQUENTIAL HEURISTIC

The heuristic is quite effective but various types of defects can be noted. Below are three configurations for which the heuristic does not produce optimal results. We use the total value objective function for toy examples.

5.1. Sequential Computation Defect

If we consider the table

40	16	4	4	16
16	10 s	2	0	4
4	2	2	0	0
4	0	0	2	2
16	4	0	2	10 s

in which the two cells with value 10 are sensitive and require protection. The sequential heuristic will yield the table

40	16	4	4	16
16	10 s	2 c	0	4
4	2 c	2 c	0	0
4	0	0	2 c	2 c
16	4	0	2 c	10 s

with 6 complements with a total value of 12. Rather we might have sought

40	16	4	4	16
16	10 s	2	0	4 c
4	2	2	0	0
4	0	0	2	2
16	4 c	0	2	10 s

that has 2 complements with a total value of 8. Both sensitive cells are protected. The value of the complements is greater than in the sequential steps. This example illustrates the need for protecting more than one cell at each step.

5.2. Computation Order Defect

If we consider the table

100	40	40	20
25	15 s	10	0
20	10	10	0
35	15	10	10
20	0	10	10 s

in which the sensitive cell of value 15 is protected before the sensitive cell of value 10. The result will be the table

100	40	40	40
25	15 s	10 c	0
20	10 c	10 c	0
35	15	10 c	10 c
20	0	10 c	10 s

in which only one internal cell is released. If we protect the cells in the other order we will obtain the table

100	40	40	40
25	15 s	10 c	0
20	10	10	0
35	15 c	10 c	10 c
20	0	10 c	10 s

in which two internal cells are released. The second step was able to use a complement from the first step in the revised sequence of computation.

5.3. Knapsack Problem Defect

If we consider the table

16	10 s	4	2

in which the sensitive cell needs to be protected by cells of size 2, we obtain the table

16	10 s (2)	4	2 c (2)

If the sensitive cell needs protection by cells of size 4, we obtain the table

16	10 s (4)	4 c (2)	2 c (2)

If the sensitive cell needs protection by cells of size 6, we obtain the table

16	10 s (6)	4 c (4)	2 c (2)

However, for the intermediate case we could also have obtained the table

16	10 s (4)	4 c (4)	2

This type of defect arises in continuous based optimization which may partially use the larger complementary cells.

6. RESOLVING THE DEFECTS

With instructive examples of the various forms of defects in the greedy sequential heuristic, we may develop refinements in the computational process to address the defects.

6.1. Knapsack Problem Defect

The presence of knapsack problems within cell suppression problems has long been noted. They form the theoretical basis for the proofs that the cell suppression problem is a hard problem within the definitions of complexity theory. There are well known methods for solving knapsack problems. For cell suppression those methods translate into integer programming methods for obtaining solutions of the single cell protection problem.

If we consider the table

16	10s	3	2	1

in which the sensitive cell needs to be protected by cells of size 4 we get the table

16	10 s (4)	3 c (1)	2 c (2)	1 c (1)

A continuous optimizer would first use the cell of size 1, then the cell of size 2 and finally the cell of size 3. This is the simplest approximation to a knapsack solution in which one keeps adding the small items until the threshold is exceeded, for our variant where we want the smallest summation that exceeds the specified value. We can improve this by reversing the order of inclusion to eliminate some of the smallest cells and obtain the table

16	10 s (4)	3 c (3)	2 c (1)	1

To achieve this in a cell suppression problem we would need to do several things to convert the standard problem into this two stage problem. We would treat all of the cells released by the initial solution as fixed for this stage. The objective function would have to give preference to the largest complements, as distinct from the earlier solution where it gave preference to smaller complements. The coefficients in the objective function could be $log(1+x)/(1+x)$ or perhaps $1/x$. The reduction of a cell suppression method to the implied knapsack method often provides useful insights into the cell suppression method.

The table

16	10 s (4)	3 c (3)	2	1 c (1)

would be an even better solution but would require an integer based optimizer to find such a solution. There is ample opportunity to adapt integer optimization methods to the cell suppression problem and solve this knapsack defect more fully than the continuous based optimization methods.

6.2. Computation Order Defect

When we watch the greedy sequential heuristic in operation, we see some large complements arising in the later steps of the computation. We would say that it has not been able to look ahead to discover these large complements.

To improve the look ahead capability we can redo the computation except on the second trial we already know the large complements from the later stages. We can make the heuristic assumption that they will be unchanged under other orderings of the steps. They will be rediscovered so we might as well supply them at the beginning of the second trial. This can be readily implemented by a first phase that determines a suppression pattern including the large complements and a second phase that starts with the large complements and determines a suppression pattern.

We could alternately say that the method of ordering the sequential steps is the defect. The ordering should be based on the size of the complements that each sensitive cell requires. This information is not initially available but can be constructed. We would treat each sensitive cell as an independent problem in a first phase used to determine the size of the complements required. For the second phase we would use the usual greedy sequential heuristic computation with the revised ordering based on the size of the complements. This can be readily implemented although the number of independent problems in the first phase will be larger than the number of steps in the greedy sequential heuristic as no sensitive cells will be protected in passing with the saving in computational cost that results.

6.3. Sequential Computation Defect

To avoid the sequential computation defect we would need to do all of the single cell protection cases in a single step. We would keep track of the extreme case over the concurrent cases. The extreme case would serve as an envelop for the individual cases which protect individual sensitive cells. The envelop would not satisfy the defining relations although the individual cases would. The combining rule which gives the envelop is to take the minimum and maximum. This can readily be done as the inequalities that represent this operation are just $-x_i \leq x_{j,i} \leq x_i$ where x_i is a variable in the envelop and $x_{j,i}$ is the corresponding variable in individual case j. Only the envelop variables would appear in the objective function. Sensitive and presuppressed cells would not require envelop variables as their effect on the envelop is already known although they may be needed for other purposes. The number of variables goes up quickly as we need to represent the envelop, the individual cases and the slack variables with respect to the envelop. In earlier *mainframe* computing eras this would not have been contemplated as there was not enough memory available for the many variables. Contemporary *workstations* have adequate memory to allow this to be easily done for toy problems. Small production problems can also be done for important publications and to gain insight into the importance of the sequential computation defect.

When a table has a simple structure, the defining relationships correspond to a network structure. Many algorithms for cell suppression are based on this simplifying assumption as it permits the use of the network simplex algorithm for the underlying optimization in the greedy sequential heuristic. The network simplex algorithm has a particularly elegant theory and permits very efficient implementations. When we combine several copies of a network, as we do here, with a maximum operator the result is no longer a network. Even for the simple example given above, the combined equation system will not correspond to a network.

In terms of the Russian doll metaphor, the envelop is the outer Russian doll that we are seeking to design. The inner bean bags are the individual cases with only a few actually preventing the outer doll from shrinking further.

Various techniques are used in operations research to exploit the fact that only a few of the equations are active. These techniques have many adaptations to identify the active equations that are required to define the solutions. Freschetti and Salazar (1998) have developed a method motivated by the Bender's decomposition for mixed integer programs with the addition of various speedups. They use an objective function of weighted integer variables. One of their speedups is a version of the common logical heuristic of requiring at least two suppressions in an equation when any suppressions are present. They do not note the connection.

If we look at defining equation j from $D\ x = 0$ we will have $\Sigma_i\ \sigma_{j,i}\ x_{j,i} = 0$ where $\sigma_{j,i}$ are the plus or minus coefficients. If some $x_{j,i}$ is not zero, then there will be at least one other $x_{j,i}$ which is also not zero, and similarly for the corresponding envelop variables x_i. For the envelop variables we will have $x_k \leq \Sigma_{i \neq k}\ x_i$ where k may be any of the indices in the equation. This is the numerical version of the logical requirement that there be either no suppressions or two or more suppressions in every equation. The derivation is a repeated application of the triangle inequality for absolute values after the original defining equation has been rearranged so that the chosen term equals a combination of the other terms followed by recalling the definition of the envelop variables. Each of the original defining equations will correspond to several of these relationships in the envelop variables. We could treat them as equations in only the envelop variables with none of the variables $x_{j,i}$ present. The logical requirement is known to be necessary but not sufficient for the prevention of disclosures. The numerical version has the same problem as can be shown by the standard examples. We have two possible resolutions to this problem.

We may view a solution with only the envelop variables as an initial approximation, or good starting value, for the greedy sequential heuristic. By the time the greedy sequential heuristic completes it will have protected every sensitive cell by adding any required complements. There may be some spurious complements so a knapsack problem clean up phase would be indicated.

We may also view each residual disclosure in the envelop variables only solution as an indication of a missing equation. This missing equation will be obtained by combining the existing equations in ways which effect the maximization for the envelop variable differently than those already present. The combination may be found using linear programming duality theory for each disclosure. A group of new equations in the envelop variables may be added to those already known and a new solution attempted. This is the basis of the Bender's decomposition method of Freschetti and Salazar. Use of only the active equations, and other operations research improvements, are also part Freschetti and Salazar's method.

7. COMPUTATIONAL EXPERIENCE

The knapsack problem defect has been addressed by the two pass strategy with revised weights in both CONFID and the ACS Suite of software. The second, or clean up, pass releases many complements. The use of two passes is the default operational mode for the ACS Suite (Sande 1999).

The various heuristics have been exercised with the EIA test data. The reordering and look ahead strategies provide incremental improvements. The improvements from the clean up of the greedy sequential heuristic deserve to be described as more that just incremental.

By monitoring the suppressions in the internal cells and the marginal cells separately, we can gain some insight into the effects of the varying objective functions. The number of cells suppressed increases as the objective changes from the count of suppressed cells, to the of suppressed cells entropy and then to the sum of suppressed values. This masks the effect that the number of internal cells suppressed is increasing while the number of marginal cells suppressed fluctuates. The total value of the suppressed cells is decreasing with the value of the suppressed marginal cells showing the greatest decrease while the value of the suppressed internal cells declines and fluctuates. The entropy of the both the internal and marginal suppressed cells is least for the entropy objective with the internal cells more closely matched by the count objective and the marginal cells more closely matched by the value objective. The overall behaviors of the objective functions match their descriptions although the descriptions might not have suggested the form of the balancing between the suppression of internal and marginal cells.

The simultaneous computation is more expensive than the greedy sequential heuristic even when it is with the envelop variables only. When the same number of cells are present in a two dimensional structure and a three dimensional structure the computational cost is markedly different, with the three dimensional much more expensive. This major difference had not been observed with the greedy sequential heuristic. For the sequential greedy heuristic a three dimensional structure will usually have a slightly higher count of equations than the same number of cells in a two dimensional structure. The equation count difference is greater in the envelop variables only problems but the time differences are even greater. Freschetti and Salazar also report increased computational costs when they move away from two dimensional examples.

	Objective Function Coefficients		
	Value	Entropy	Count
Value Measure			
Internal	12 42687	11 74618	64 46070
Marginal	3 18507	25 94011	118 53544
Total	15 61194	37 68629	182 99614
Entropy Measure			
Internal	3683.1	2948.1	3202.3
Marginal	725.5	673.9	969.0
Total	4408.6	3622.0	4171.3
Count Measure			
Internal	561	439	401
Marginal	102	78	93
Total	663	517	494

8. CONCLUSIONS

The greedy sequential heuristic is quite effective. It arises naturally and has been rediscovered repeatedly. Various defects in it can be recognized as having understandable sources that lead to refinements in the basic algorithm. These refinements produce improved suppression patterns.

The Russian doll metaphor illustrates the way in which the greedy sequential heuristic, with extra passes, naturally approximates the simultaneous computation.

9. ACKNOWLEDGMENTS

The test micro data available from the EIA web site has been used for the testing reported here. This test data set is defined by the test data provided for the larger example of use of the USBC cell suppression software.

10. REFERENCES

Fischetti, M. and J.-J. Salazar-Gonzalez (1998), "Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints," manuscript.

Kirkendall, N. and G. Sande (1998), "Comparison of Systems Implementing Automated Cell Suppression for Economic Statistics," *Journal of Official Statistics*, **14**, pp 513-535.

Robertson, D. (1993), "Cell Suppression at Statistics Canada," *Proceedings of the 1993 Annual Research Conference*, Bureau of the Census, pp 107-131.

Sande, G. (1984), "Automated Cell Suppression to Preserve Confidentiality of Business Statistics," *Statistical Journal of the United Nations ECE*, **2**, pp 33-41.

Sande, G. (1999), "Structure of the ACS Automated Cell Suppression System," *Statistical Data Confidentiality, Proceedings of the Joint Eurostat / UN-ECE Work Session on Statistical Confidentiality*, pp 105-121, ISBN 92-828-7747-7.